Available online at www.sciencedirect.com
ScienceDirect

Neural Networks 20 (2007) 454–461

Neural
Networks

www.elsevier.com/locate/neunet

2007 Special Issue

Reducing uncertainties in neural network Jacobians and improving accuracy of neural network emulations with NN ensemble approaches

Vladimir M. Krasnopolsky*

Science Applications International Corporation at Environmental Modeling Center, National Centers for Environmental Prediction, National Oceanic and Atmospheric Administration, MD, USA
Earth System Science Interdisciplinary Center, University of Maryland, MD, USA

Abstract

A new application of the NN ensemble technique to improve the accuracy (reduce uncertainty) of NN emulation Jacobians is presented. It is shown that the introduced ensemble technique can be successfully applied to significantly reduce uncertainties in NN emulation Jacobians and to reach the accuracy of NN Jacobian calculations that is sufficient for the use in data assimilation systems. An NN ensemble approach is also applied to improve the accuracy of NN emulations themselves. Two ensembles linear (or conservative) and nonlinear (uses an additional averaging NN to calculate the ensemble average) were introduced and compared. The ensemble approaches: (a) significantly reduce the systematic and random error in NN emulation Jacobian, (b) significantly reduce the magnitudes of the extreme outliers and, (c) in general, significantly reduce the number of larger errors. It is also shown that the nonlinear ensemble is able to account for nonlinear correlations between ensemble members and to improve significantly the accuracy of the NN emulation as compared to the linear conservative ensemble in terms of systematic (bias), random, and larger errors.

Keywords: Neural networks; Ensembles; Numerical modeling; Climate; Weather; Data assimilation

1. Introduction

The multilayer perceptron (MLP) neural network (NN) is a generic analytical nonlinear approximation or a model for nonlinear (continuous) mappings (Funahashi, 1989). The simplest MLP NN uses for the approximation of mappings a family of functions like:

$$y_q = a_{q0} + \sum_{j=1}^{k} a_{qj} \cdot \tanh\left(b_{j0} + \sum_{i=1}^{n} b_{ji} \cdot x_i\right); \qquad q = 1, 2, \ldots, m \tag{1}$$

where x_i and y_q are components of the input and output vectors respectively, a and b are matrixes of fitting parameters (NN weights and biases).

A mapping, which NN (1) approximates, (i.e., the target mapping) can be symbolically written as a mapping between two vectors, X (input vector) and Y (output vector):

$$Y = M(X); \quad X \in \mathfrak{R}^n, Y \in \mathfrak{R}^m. \tag{2}$$

A large number of important practical applications in geosciences may be considered mathematically as a mapping (2) (Fox-Rabinovitz, Krasnopolsky, & Belochitski, 2006; Hsieh, 2001; Krasnopolsky, Breaker, & Gemmill, 1995; Krasnopolsky, Gemmill, & Breaker, 1999; Krasnopolsky, Chalikov, & Tolman, 2002; Krasnopolsky, Fox-Rabinovitz, & Chalikov, 2005; Krasnopolsky & Fox-Rabinovitz, 2006; Krasnopolsky et al., 2006). The regular NN approximation technique provides an NN approximation of the mapping with sufficiently small approximation errors on the training set. However, a satisfactory generalization (interpolation) capability of the NN approximation is not guaranteed without additional constraints. In many applications the necessity arises to improve the generalization (interpolation) accuracy of developed NN approximations. NN emulation techniques have

* Corresponding address: Environmental Modeling Center, 5200 Auth Rd., Camp Springs, MD 20746-4304, USA. Tel.: +1 301 763 8000x7262; fax: +1 301 763 8545.
E-mail address: vladimir.krasnopolsky@noaa.gov.

0893-6080/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2007.04.008

been developed to emulate mappings with high approximation and interpolation accuracy (e.g., Krasnopolsky et al. (2005); see also Section 2 of this paper).

In some applications, developed NN emulations of mappings (2) are used for inversion (Krasnopolsky et al., 1995, 1999) in a data assimilation system (DAS) (Krasnopolsky, 1997; Krasnopolsky et al., 2006) or for error and sensitivity analysis. In all these cases not only the mapping (2) but also its first derivatives are used. It means that the NN emulation Jacobian that is a matrix of the first derivatives of the outputs of the NN emulation (1) over its inputs, $\left\{\frac{\partial y_q}{\partial x_i}\right\}_{i=1,\ldots,n}^{q=1,\ldots,m}$, has to be calculated. From a technical point of view, the calculation of the Jacobian is almost trivial. It is performed by an analytical differentiation of Eq. (1). However, from a theoretical point of view, the inference of the NN Jacobian is an ill-posed problem (Vapnik, 1995) which leads in practice to significant uncertainties in calculated NN Jacobians (Aires, Schmitt, Chedin, & Scott, 1999; Aires, Prigent, & Rossow, 2004; Chevallier & Mahfouf, 2001).

For applications that require an explicit calculation of the NN Jacobian, several solutions have been offered and investigated to reduce the NN Jacobian uncertainties. First, the Jacobian can be trained as a separate additional NN (Krasnopolsky et al., 2002). Second, the mean Jacobian can be calculated and used (Chevallier & Mahfouf, 2001). Third, regularization techniques like "weight smoothing" (Aires et al., 1999) or the technique based on a principal component decomposition (Aires et al., 2004) can be used to stabilize the Jacobians. Fourth, the Jacobian can be trained, i.e. included as actual additional outputs in the NN and in the training data set. To do this, the error (or cost) function, which is minimized in the process of NN training, should be modified to accommodate the Jacobian; in other words, the Euclidian norm, which is usually used for calculating the error function, should be changed to the first order Sobolev's norm. Actually, Hornik, Stinchcombe, and White (1990) showed that the function of the Sobolev's space can be approximated with all their derivatives. This and other similar theoretical results are very important because they prove the existence of the approximation.

In this paper we introduce a new NN ensemble approach to reduce uncertainties in calculated NN emulation Jacobians. NN ensemble approaches have been introduced by many authors (Barai & Reich, 1999; Hashem, 1997; Maclin & Shavlik, 1995; Naftaly, Intrator, & Horn, 1997; Opitz & Maclin, 1999; Sharkey, 1996). They were used to improve NN classification, NN approximation and NN generalization abilities. To the best of our knowledge, this work is the first one that introduces an application of an NN ensemble technique for reducing uncertainties of NN emulation Jacobians.

In this paper we also compare a linear or conservative ensemble (Barai & Reich, 1999), where simple (with equal weights for all members) averaging of the members provides the ensemble mean and other statistics, with a nonlinear ensemble where an averaging NN is introduced that takes into account nonlinear correlations between ensemble members. This averaging NN, given ensemble members as inputs, generates a nonlinear ensemble average.

In Section 2 of this paper we define NN emulation of complex mappings; we also discuss a generic approach to use NN ensembles to improve the accuracy of the NN emulation Jacobian and the NN emulation. In Section 3 we illustrate these approaches using as a test bed a particular practical application that is described in detail in Krasnopolsky et al. (2006), an NN emulation for the ocean surface elevation mapping in DAS of an ocean numerical model. Conclusions are presented in Section 4.

2. Background: NN emulations and NN ensemble approaches to NN emulations and NN emulation Jacobians

2.1. NN emulations of complex mappings

In this paper, we use the terms an *emulating NN* or a *NN emulation* of the mapping (2). An emulating NN (1) provides a functional emulation of the target mapping (2) that implies a small approximation error for the training set and smooth and accurate interpolation between training set data points inside the mapping domain D. The term "emulation" is introduced to distinguish between these NNs and approximating NNs or NN approximations that guarantee small approximation error for the training set only.

When an emulating NN is developed, in addition to the criterion of small approximation error at least three other criteria are used: (i) the NN complexity (proportional to the number k of hidden neurons when other topological parameters are fixed) is controlled and restricted to a minimal level sufficient for good approximation and interpolation; (ii) independent validation and test data sets are used; the validation set is used in the process of training to control overfitting and the test set is used after the training to evaluate interpolation accuracy; (iii) redundant training set (additional redundant data points are added "in-between" training data points sufficient for a good approximation) is used for improving the NN interpolation abilities.

2.2. Multiple NN emulation solutions and ensemble approach

As a nonlinear model or nonlinear approximation of the mapping (2), the NN approximation problem allows for multiple solutions or for multiple NN emulations (1) for the same mapping (2). Existence of multiple solutions is a common property of nonlinear models, or of nonlinear approximations. These models have nonlinear parameters that may be changed to generate solutions, which may be close in terms of satisfying a particular criterion (e.g., approximation error) used for obtaining the solutions. For example, the same mapping (2) can be emulated with NNs (1) with different numbers of hidden neurons, with different weights (resulting from the NN training with different initializations that lead to different local minima of the cost function), different partitions of the training set, etc. At the same time, these multiple models (NNs) may be different in terms of other criteria providing complementary information about the target mapping (2). The availability of multiple solutions may lead to some inconveniences like the necessity to introduce an additional step that is to use additional criteria to select a single, optimal model or problems like uncertainties or multiple solutions for the NN emulation Jacobian. The existence of multiple and significantly different solutions for the NN Jacobian is a consequence of the fact that the statistical inference of the NN Jacobian is an ill-posed problem (Vapnik, 1995).

On the positive side, availability of multiple models (multiple NN emulations and NN emulation Jacobians), providing complementary information about the target mapping (2) and its Jacobian, opens an opportunity to use ensemble approaches. Ensemble approaches allow for integrating complementary information, contained in the ensemble members, into an ensemble that "knows" more about or represents the mapping (2) and its Jacobian better than each of the individual ensemble members (a particular NN emulation, or a particular NN emulation Jacobian).

An ensemble of NNs consists of a set of members, that are individually trained NNs. They are combined when applied to a new input data to improve the generalization (interpolation) ability. Previous research showed that an ensemble is often more accurate than any or most of the individual members of the ensemble (Bishop, 1995). Previous research also suggests that any mechanism that causes some randomness or perturbation in the formation of NN ensemble members can be used to form an NN ensemble (Opitz & Maclin, 1999). For example, ensemble members can be created by training different members: (a) on different subsets of the training set (Opitz & Maclin, 1999); (b) on different subdomains of the training domain; (c) using NNs with different topologies (e.g., different number of hidden neurons) (Hashem, 1997); (d) using NNs with the same architecture but with different initial conditions for NN weights (Maclin & Shavlik, 1995; Sharkey, 1996).

In the context of our application, i.e. approximation of a complex mapping (2), the members of the ensemble are separately trained NNs which provide emulations for the target mapping with slightly different approximation and interpolation accuracies. Because of the properties of the NN emulations described in Section 2.1, we can expect that these emulations do not oscillate strongly between the training set data points. It means that the spread of the emulation accuracy and, what is even more important, the spread between different solutions for the NN Jacobian (the Jacobian uncertainties) are limited. Thus, we can expect that the ensemble average will provide a better approximation and interpolation than its individual members.

2.3. *Conservative ensemble vs. nonlinear NN ensemble*

Different ways of combining NN ensemble members into the ensemble have been developed and investigated (Naftaly et al., 1997). In this work, we start from using a conservative ensemble (Barai & Reich, 1999) where simple (with equal weights for all members) averaging of the members provides the ensemble mean and other statistics. The conservative ensemble or its modifications (linear averaging with non-equal weights) are mostly popular in applications. However, these approaches cannot account for possible nonlinear correlations between ensemble members. In this paper, we also use a nonlinear averaging approach, an NN ensemble averaging, with an averaging NN that takes into account nonlinear correlations between ensemble members. The averaging NN is trained given ensemble members as inputs to generate a nonlinear ensemble average. In the following section, it is shown that this approach may in addition (as compared to the conservative ensemble) significantly reduce the random and systematic components of approximation and interpolation errors.

3. Applications of the NN ensemble approach

Here we introduce applications of the NN ensemble approach in the context of an oceanic data assimilation problem described in more detail by Krasnopolsky et al. (2006). In this work NN was used to emulate functional nonlinear dependencies and mappings between ocean state variables. These mappings are implicitly embedded in the solutions of highly nonlinear coupled partial differential equations of an ocean dynamical model and, therefore, in numerical outputs of this model.

In particular, in a layered ocean model, where the model physics is usually approximated by 1D parameterizations depending on the depth only, the sea surface height (SSH or η) signal at a particular horizontal location depends mainly on the vertical disposition of the horizontal layers in a vertical column at this particular horizontal location. Therefore, this dependence, after emulating it with NN, can be written as

$$\eta_{\mathrm{NN}} = \phi_{\mathrm{NN}}(X), \tag{3}$$

where ϕ_{NN} is an NN and X is a vector that represents a complete set of ocean state variables at a particular location, which determines SSH at this horizontal location. Thus, the vector X depends on the vertical model coordinate only, and the mapping (3) does not depend on time and horizontal location of the vector X explicitly. The mapping (3), as many parameterizations in atmospheric and oceanic model physics, works in the space of the ocean states and depends on time and horizontal location indirectly or implicitly through changes in the vector of ocean state variables X.

In this paper we work with a version of Real Time Ocean Forecast System (RTOFS) (Atlantic) — the operational model of the Atlantic Ocean running at the National Centers for Environmental Predictions. In this model the vector X was selected as $X = \{I, \theta, z_{\mathrm{mix}}\}$, where I is the vector of interfaces, the vertical coordinates used in RTOFS (Atlantic) (Krasnopolsky et al., 2006), θ is the vector of potential temperature, and z_{mix} is the depth of the ocean mixed layer (a total of 50 variables).

An analytical NN emulation (3) for the relationship between model state variables, X, and sea surface height, η, was derived using the simulated model fields for η, I, θ, and z_{mix} which were treated as error-free data (Krasnopolsky et al., 2006). A simulation that covers almost two years (from Julian day 303, 2002 to 291, 2004) was used to create training, validation, and test data sets. The periods covered by these data

Table 1
Periods covered by training, validation and test data sets and their sizes

Set	Beginning date (Julian day, year)	End date (Julian day, year)	Size, N (number of records)
Training	303, 2002	52, 2004	563, 259
Validation	303, 2002	52, 2004	563, 259
Test	53, 2004	291, 2004	563, 259

sets and their sizes are shown in Table 1. Each data set consists of records $\{\eta_p, X_p\}_{p=1,\ldots,N}$ collocated in space and time and uniformly distributed over the model domain. The model that we use is a high resolution one; its horizontal resolution is about $1/3^\circ$. The dimensionality of the grid that covers the model domain is too high to include all grid points at each time step in our data sets. For the training and validation data sets, at each time step two different non-overlapping subsets of horizontal grid points were selected. Thus the training and validation sets are independent because there is no single data record in them that belongs to the same time and location. The test set is completely independent of the training and validation data sets because it is separated in time (see Table 1).

We use the following setup to create an ensemble of NNs. The complexity of the NN emulation (3) was limited; only three hidden neurons were allowed (see Section 2.1). Then ten NN emulations (3) with the same number of the hidden neurons (three) were trained using differently perturbed initial conditions for the NN weights. As a result, an NN ensemble that consists of ten members, ten NN emulations with identical architecture (50 inputs, 3 hidden layers, and 1 output) but different weights and different approximation accuracies has been created. We varied the number of the ensemble members from eight to twelve; no significant changes in the results presented below for the ensemble of ten members were found.

3.1. NN ensembles for reducing uncertainties of the NN Jacobian

The NN emulation (3) can be used in the ocean DAS to enhance assimilating SSH and to improve the propagation of the surface SSH signal to other vertical levels and other variables during the data assimilation cycle. In the ocean DAS the increment of the SSH, $\Delta\eta$, is calculated using the NN Jacobian $\{\frac{\partial\phi_{\mathrm{NN}}}{\partial X_i}\}_{i=l,\ldots,n}$,

$$\Delta\eta_{\mathrm{NN}} = \sum_{i=1}^{n} \left.\frac{\partial\phi_{\mathrm{NN}}}{\partial X_i}\right|_{X=X_0} \cdot \Delta X_i, \tag{4}$$

where ΔX_i are increments of state variables, X_0 is an initial value of state variables and n is the dimensionality of the vector X (the number of inputs of the NN emulation (3)). Then the calculated $\Delta\eta_{\mathrm{NN}}$ is compared to the observed $\Delta\eta_{\mathrm{obs}}$ and the difference is used to adjust ΔX in a variational DAS.

The quality of the single NN Jacobian may not be sufficient for the use in DAS applications. However, an ensemble approach can be used to improve the NN Jacobian calculations. The NN ensemble described above was used here to create an ensemble of ten NN Jacobians $\{\frac{\partial\phi^j_{\mathrm{NN}}}{\partial X_i}\}^{j=l,\ldots,\mathrm{ens}}_{i=l,\ldots,n}$, where ens $= 10$ is the number of the ensemble members. Then the ensemble average Jacobian has been calculated,

$$\overline{\frac{\partial\phi_{\mathrm{NN}}}{\partial X_i}} = \frac{1}{\mathrm{ens}} \sum_{j=1}^{\mathrm{ens}} \frac{\partial\phi^j_{\mathrm{NN}}}{\partial X_i}, \quad i = 1, \ldots, n. \tag{5}$$

Now, Eq. (4) was used to calculate $\Delta\eta_{\mathrm{NN}}$ using each ensemble member Jacobian and the ensemble average Jacobian (5). These values of $\Delta\eta_{\mathrm{NN}}$ were compared with exact $\Delta\eta$ known from the model simulation.

The comparison technique was applied to the independent test set (see Table 1). The use of the ensemble average Jacobian (5) reduces the RMS error on the test set by 1–4 cm as compared to the single member Jacobian (4). The comparison technique was also applied to various days of the simulation selected from the test set. The results obtained for the different days of the simulation are very similar to those presented here for the last day of the entire model simulation. This day was selected because it is separated by the time interval of about 8 months from the last day of simulation used for NNs training and validation (see Table 1). Fields generated by the model at 00Z were used to create inputs, X, for the NN emulation Jacobians. Then the NN emulation Jacobian ensemble members were applied (4) over the entire domain (with coastal areas excluded) to generate an ensemble of 2D fields of $\Delta\eta^j_{\mathrm{NN}}$. Then $\overline{\Delta\eta_{\mathrm{NN}}}$ was calculated using the ensemble average Jacobian (5) in (4). Also a nondimensional distance in the model state space between vectors X_0 and $X = X_0 + \Delta X$ was introduced,

$$S = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\Delta X_i}{X_0^i}\right)^2}. \tag{6}$$

These fields were compared with the corresponding field of SSH, η, generated by the model. We performed multiple case studies for particular locations inside the RTOFS (Atlantic) domain to better illustrate the results. Results of a typical case study that uses the test data are presented in Figs. 1–4.

Fig. 1 shows the location of the cross section (white horizontal line) inside the model domain for this case; white dot shows the position of X_0. Starting from this position we moved left and right grid point by grid point using X values at these grid points to calculate ΔX and the nondimensional distance in the model state space, S. These values of ΔX were used in (4) to calculate $\Delta\eta$.

An envelope of thin solid lines depicted in Fig. 2 shows $\Delta\eta$ calculated using (4) and the different NN ensemble member Jacobians. Fig. 2 also shows the exact $\Delta\eta$ calculated from the model (thick solid line) and $\Delta\eta$ calculated using the

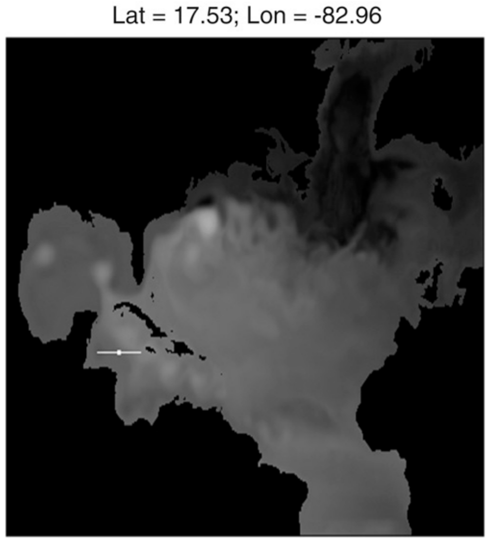

Fig. 1. The location of the cross section (white horizontal line) inside the RTOFS (Atlantic) model domain; white dot shows the position of X_0.

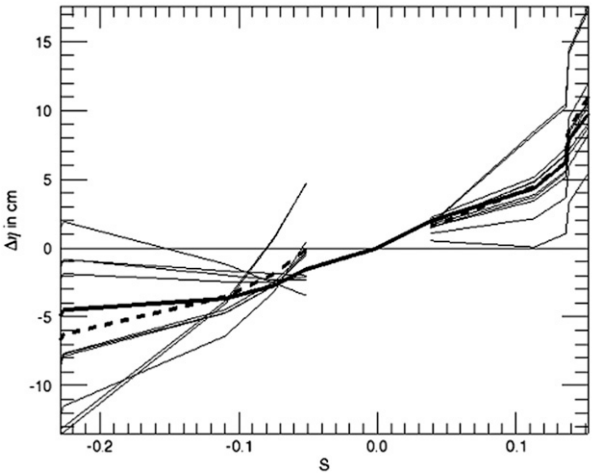

Fig. 2. An envelope of thin solid lines shows $\Delta\eta$ calculated using (4) and the different NN ensemble member Jacobians. The width of the envelope illustrates the Jacobian uncertainties. Thick solid line shows the exact $\Delta\eta$ calculated from the model, and thick dashed line the $\Delta\eta$ calculated using the ensemble average Jacobian (5). $\Delta\eta$ is shown vs. the nondimensional distance in the model state space, S (6).

ensemble average Jacobian (5) (thick dashed line). $\Delta\eta$ is shown vs. the distance in the model state space, S. This figure demonstrates that the use of the ensemble average improves the NN Jacobian very significantly. The larger the distances S, the more significant is the reduction of the Jacobian uncertainties when using ensemble average.

Fig. 3 shows the systematic error (bias) and the random error (error standard deviation) for $\Delta\eta$ calculated along the path shown in Fig. 1 using (4). The asterisks show errors when the ensemble member Jacobians were used in (4), the cross shows the errors in the case when the ensemble average Jacobian (5) was used. The ensemble bias is equal to the mean bias of the members as it can be expected using this simple method of calculating the ensemble average. This figure also shows that in the case of Jacobian the ensemble approach is very effective in reducing random errors; it shows that ensemble random error (1.1 cm) is less than the random error of any of the ensemble members. The reduction in systematic (∼90%) and random (∼65%) errors with respect to the maximum single member errors is very significant.

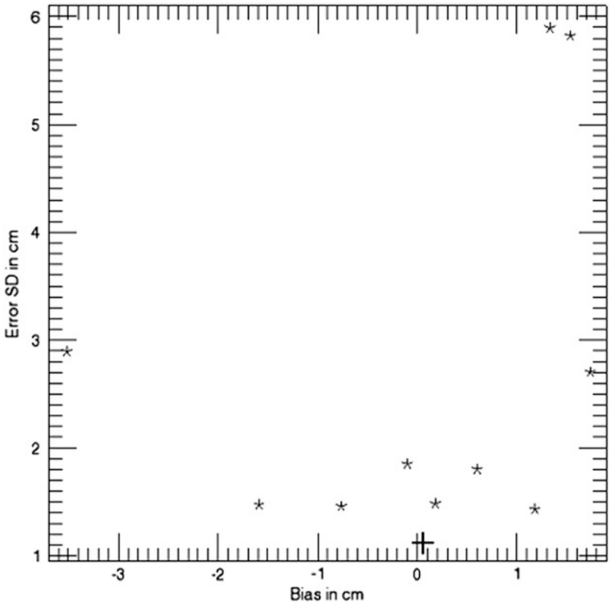

Fig. 3. The systematic error (bias) and the random error (error standard deviation, SD) for $\Delta\eta$ calculated along the path shown in Fig. 1 using (4). The asterisks correspond to errors when the ensemble member Jacobians were used in (4), the cross corresponds to the case when the ensemble average Jacobian (5) was used.

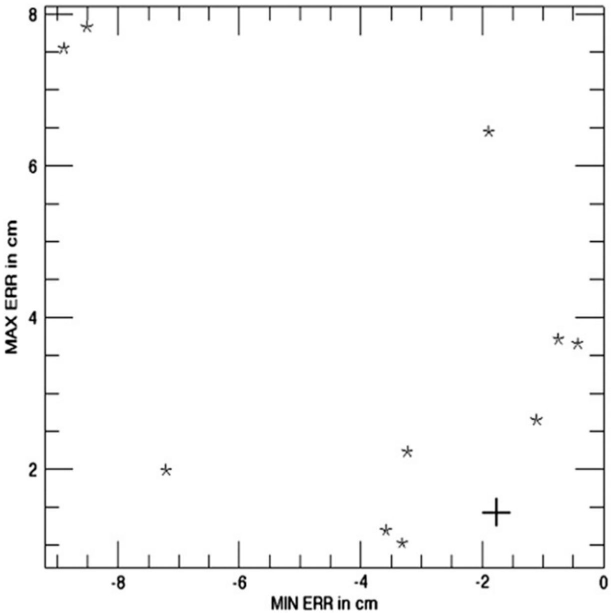

Fig. 4. The minimum and maximum errors along the path shown in Fig. 1. The asterisks correspond to errors when the ensemble member Jacobians were used in (4), the cross corresponds to the case when the ensemble average Jacobian (5) was used.

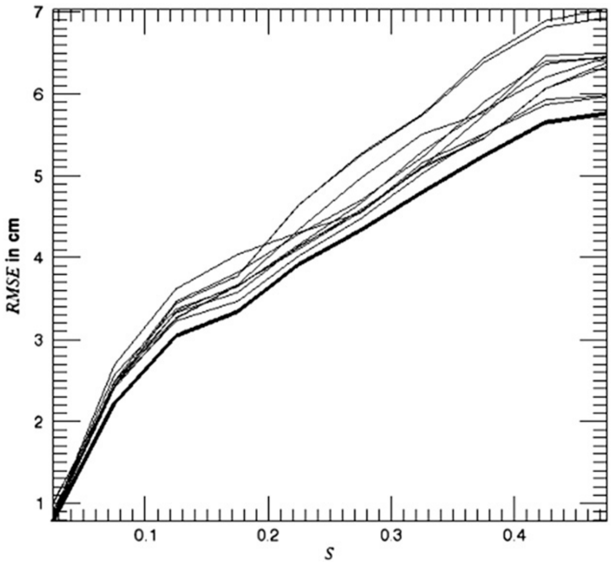

Fig. 5. Errors in $\Delta\eta$ as functions (binned and averaged in each bin) of nondimensional distance S over entire model domain. Thin lines correspond to the ensemble members and the thick line shows the ensemble average result.

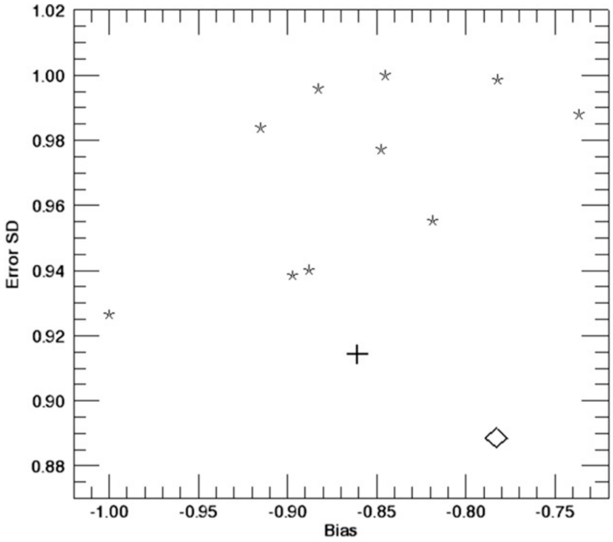

Fig. 6. Vertical axis shows the random part of the emulation error (the standard deviation of the error) and the horizontal axis—the systematic error (mean error or bias). Both errors are normalized to the corresponding maximum (among the ten ensemble members) error. Each ensemble member is represented in the figure by a star, the conservative ensemble average—by the cross, and the nonlinear ensemble using the averaging NN—by diamond.

Fig. 4 shows minimum and maximum errors or statistics for extreme outliers along the same cross section. When each ensemble member NN Jacobian is applied in (4), for each particular input vector the NN produces an error. Among all these errors there exist one largest negative (or minimum) error and one largest positive (or maximum) error or two extreme outliers that demonstrate the worst case behavior (scenario) that we can expect from this particular NN emulation. These two extreme outliers (negative and positive) are presented as a star for each NN member in the figure. The ensemble average Jacobian (5), when used in (4), also generates such two extreme outliers that are presented as the cross in the figure. The figure shows that the NN ensemble approach is also an effective tool in reducing (∼4 times) large errors in NN Jacobians.

Next we applied the same procedure at all grid points of the model domain. The errors have been calculated along numerous paths (horizontal and vertical) all over the model domain. Fig. 5 shows RMS error in $\Delta\eta$ as a function (binned and averaged in each bin) of nondimensional distance S over the entire domain. Thin lines correspond to the ensemble members (an envelope of thin solid lines illustrates the Jacobian uncertainties) and the thick line shows the ensemble result. The ensemble significantly improves statistics at all considered distances S. The ensemble average is always better than the best ensemble member.

To better understand the magnitudes of errors presented in this and the next section, these magnitudes should be compared with the errors in observed satellite data $\Delta\eta_{\mathrm{obs}}$ assimilated in the oceanic DAS (Krasnopolsky et al., 2006). The accuracy of the observed data is about 5 cm. It means that our NN emulation (3) and ensemble technique allow us to reduce the Jacobian uncertainties and produce the ensemble Jacobian (5) that has accuracy comparable with that of the observed data and is sufficiently accurate to be used in ocean DASs.

3.2. Ensemble approach to improve emulation accuracy; linear and nonlinear ensembles

Here we apply the NN ensemble approach in a more traditional mode to improve the accuracy of the NN emulation (3) (see also Fox-Rabinovitz et al. (2006)). After the NN ensemble (see above) was created, each NN member (that is a particular realization of the NN emulation (3)) was applied to the test set and the error statistics for each NN member was calculated and plotted in Fig. 6. The vertical axis of the figure shows the random part of the approximation error (the standard deviation of the error) and the horizontal axis—the absolute value of the systematic error (bias). Both errors are normalized to the corresponding maximum member error (maximum member bias or maximum member error standard deviation). Each ensemble member is presented by a star in this figure. The figure illustrates the spread of the ensemble members; it is significant. For different members, the systematic error changes about 25% and the random error about 10%.

The ensemble or the ensemble average can be produced in different ways (Barai & Reich, 1999). The first averaging approach that we use here is the simplest and linear method of the ensemble averaging—a conservative ensemble (Barai & Reich, 1999). Each of the ten members of the NN ensemble was applied to the test set record by record. Thus, for each record, for each set of inputs, ten NN outputs were produced. Then, the mean value (in a regular statistical sense) of these ten outputs was calculated and used to compare with the exact output to calculate ensemble statistics for the entire test set presented by the cross in Fig. 6. The ensemble bias is equal to the mean bias of the members as it can be expected when using this simple linear method of calculating the ensemble average. Among other things, Fig. 6 also illustrates a known fact

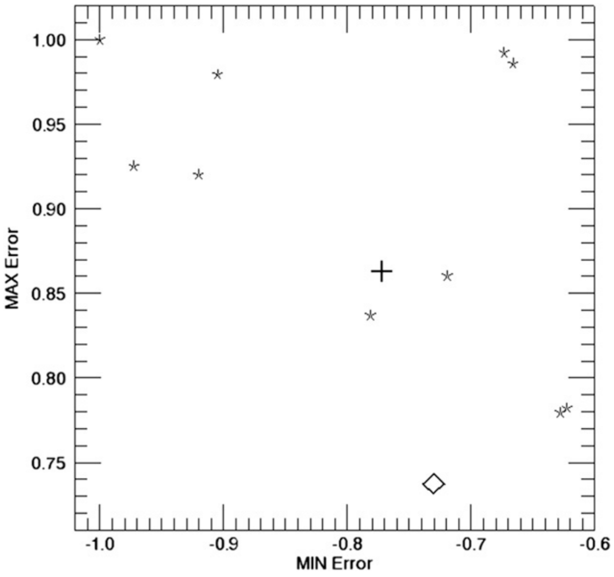

Fig. 7. Extreme outliers statistics. Vertical axis shows the largest positive (or maximum) and the horizontal axis the largest negative (or minimum) emulation error over the entire test set. Each ensemble member is represented by a star, the conservative ensemble—by the cross, and the nonlinear ensemble—by the diamond in the figure.

that ensemble approaches are very effective in reducing random errors; it shows that ensemble random error is less than the random error of any of the ensemble members. The reduction in systematic (∼15%) and random (∼9%) errors with respect to the maximum single member errors is not large but significant.

Conservative ensemble is simple; however, it is linear; it completely neglects nonlinear correlations and dependencies between ensemble members. To estimate the contribution of these nonlinear correlations and to use these correlations to improve ensemble averaging we developed a nonlinear ensemble that uses an additional averaging NN to calculate the ensemble average. The inputs of the averaging NN are constituted from the outputs of the ensemble member NNs. The number of inputs of the averaging NN is equal to the number of ensemble members multiplied by the number of outputs in a single ensemble member NN (10 in our case). It has the same outputs as a single ensemble member NN (one in our particular case). The averaging NN was trained using training and validation sets prepared on the basis of the training and validation sets used for training the ensemble member NNs. The test statistics presented here were calculated using the test set.

Fig. 6 shows statistics for nonlinear ensemble, which uses the averaging NN, with diamond. It shows that the magnitude of the nonlinear correlations between ensemble members is significant and can be successfully used to improve ensemble accuracy. Comparison of the position of the cross and the diamond in Fig. 6 shows that, as compared to the conservative ensemble, the nonlinear ensemble gives an additional improvement in bias of the order of 10%. The nonlinear ensemble bias is close to the minimum ensemble member bias. An additional improvement in the random error is a bit smaller (about 5%) but significant.

Fig. 7 shows statistics for extreme outliers. When each ensemble member NN is applied to the test set, for each record the NN produces an error. Among all these error there exist one largest negative (or minimum) error and one largest positive (or maximum) error or two extreme outliers that demonstrate the worst case behavior (scenarios) we can expect from this particular NN emulation. These two extreme outliers are presented as a star for each NN member in Fig. 7. Each ensemble also generates two such extreme outliers that are presented as a cross for the conservative ensemble and a diamond for the nonlinear ensemble in Fig. 7.

Fig. 7 shows that the NN ensemble approach is an effective tool in reducing extreme outliers (∼25%). However, a careful analysis of the figure also reveals very interesting features of the statistics presented in this figure. The distribution of stars shows a significant spread. It also demonstrates a significant clustering and correlation between extreme outliers produced by ensemble members. These facts and the position of the conservative ensemble (cross) in the figure suggest that the members of the ensemble are nonlinearly correlated. A significant improvement introduced by the nonlinear ensemble (diamond) supports this conclusion.

4. Conclusions

In this paper, we have presented a new application of the NN ensemble technique to reduce the uncertainties and improve the accuracy of the NN emulation Jacobian. We discuss the term “emulation” to stress an importance of distinguishing NN emulations from other NN models. We introduced an ensemble technique and showed that, for NN emulations, this ensemble technique can be successfully applied to significantly reduce uncertainties in NN emulation Jacobians. In the framework of an ocean data assimilation application (Krasnopolsky et al., 2006) we showed that our ensemble approach allows one to calculate the NN emulation Jacobian with the accuracy sufficient for use in the data assimilation system. The ensemble approach: (a) significantly reduces the systematic and random error in NN emulation Jacobian, (b) significantly reduces the magnitudes of the extreme outliers and, (c) in general, significantly reduces the number of larger errors.

Here and in Fox-Rabinovitz et al. (2006) we have also applied the NN ensemble approach to improve the emulation accuracy of NN emulations of complex multidimensional mappings. In particular, in Fox-Rabinovitz et al. (2006) we applied this technique to NN emulations that we developed for the long wave radiation parameterization of National Center for Atmospheric Research Community Atmospheric Model. In this paper, we applied the NN ensemble technique to a mapping developed in the framework of an ocean data assimilation application (Krasnopolsky et al., 2006). This mapping and the corresponding NN emulation relate the sea surface elevation to a vector of oceanic state variables. We introduced and compared two NN ensembles: (1) a linear or conservative ensemble estimating the ensemble average as a simple linear mean of the ensemble members, and (2) nonlinear NN ensemble that uses a special NN to estimate a nonlinear ensemble average given ensemble members. We have shown that practically all individual NN emulations that we have trained in the process of development of an optimal NN emulation, can be used, within the NN ensemble approach, for improving generalization (interpolation) ability of our NN emulations: (a) significantly reducing the systematic and random interpolation errors, (b) significantly reducing the magnitudes of the extreme outliers and, (c) in general, significantly reducing the number of larger errors. It was also shown that the nonlinear ensemble is able to account for nonlinear correlations between ensemble members and that it improves significantly the accuracy of the NN emulation as compared to the linear conservative ensemble in terms of systematic (bias), random, and larger errors.

Acknowledgments

The author thanks Dr. M.S. Fox-Rabinovitz for stimulating discussions and Dr. C. Lozano for discussions of oceanic aspects of an application used in this work. This work was supported by the NOAA CDEP-CTB Grant NA06OAR4310047.

References

Aires, F., Schmitt, M., Chedin, A., & Scott, N. (1999). The "weight smoothing" regularization of MLP for Jacobian stabilization. *IEEE Transactions on Neural Networks*, *10*, 1502–1510.

Aires, F., Prigent, C., & Rossow, W. B. (2004). Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3. Network Jacobians. *Journal of Geophysical Research*, *109*, D10305.

Barai, S. V., & Reich, Y. (1999). Ensemble modeling or selecting the best model: Many could be better than one. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *13*, 377–386.

Bishop, C. M. (1995). *Neural networks for pattern recognition* (482 pp.). Oxford, UK: Oxford University Press.

Chevallier, F., & Mahfouf, J. -F. (2001). Evaluation of the Jacobians of infrared radiation models for variational data assimilation. *Journal of Applied Meteorology*, *40*, 1445–1461.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*, 183–192.

Fox-Rabinovitz, M. S., Krasnopolsky, V. M., & Belochitski, A. (2006). Neural network ensemble approach for improving the accuracy of climate simulations that use neural network emulations of model physics. In *Proceedings of the IJCNN2006* (pp. 9321–9326). CD-ROM.

Hashem, S. (1997). Optimal linear combination of neural networks. *Neural Networks*, *10*, 599–614.

Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, *3*, 551–560.

Hsieh, W. W. (2001). Nonlinear principal component analysis by neural networks. *Tellus*, *53A*, 599–615.

Krasnopolsky, V., Breaker, L. C., & Gemmill, W. H. (1995). A neural network as a nonlinear transfer function model for retrieving surface wind speeds from the special sensor microwave imager. *Journal of Geophysical Research*, *100*, 11033–11045.

Krasnopolsky, V. (1997). A neural network-based forward model for direct assimilation of SSM/I brightness temperatures. Technical note, OMB contribution No. 140, NCEP/NOAA.

Krasnopolsky, V. M., Gemmill, W. H., & Breaker, L. C. (1999). A multiparameter empirical ocean algorithm for SSM/I retrievals. *Canadian Journal of Remote Sensing*, *25*, 486–503.

Krasnopolsky, V. M., Chalikov, D. V., & Tolman, H. L. (2002). A neural network technique to improve computational efficiency of numerical oceanic models. *Ocean Modelling*, *4*, 363–383.

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Chalikov, D. V. (2005). New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of long wave radiation in a climate model. *Monthly Weather Review*, *133*(5), 1370–1383.

Krasnopolsky, V. M., & Fox-Rabinovitz, M. S. (2006). Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, *19*, 113–121.

Krasnopolsky, V. M., et al. (2006). Using neural network to enhance assimilating sea surface height data into an ocean model. In *Proceedings of the IJCNN2006* (pp. 8732–8734). CD-ROM.

Maclin, R., & Shavlik, J. (1995). Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Proceedings of the eleventh international conference on artificial intelligence* (pp. 775–780).

Naftaly, U., Intrator, N., & Horn, D. (1997). Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems*, *8*, 283–294.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198.

Sharkey, A. J. C. (1996). On combining artificial neural nets. *Connection Science*, *8*, 299–313.

Vapnik, V. N. (1995). *The nature of statistical learning theory* (p. 189). New York: Springer.